Example 2. Analyzing time-to-event data

A. Onofri, H-P. Piepho and M. Kozak

Contents

The dataset

This simple dataset was taken from an unpublished experiment with *Lobelia erinus* L. The germination behaviour of a seed lot was tested at two different temperatures (15 and 25°C), by using one Petri dish with 100 seeds per temperature. Germinated seeds were counted every second day for 39 days, and after being counting they were removed from the Petri dishes. The experiment aimed to quantify the effect of temperature on germination velocity, by comparing the time to 50% germination (T50) at both temperatures.

Let's read the data into R, after loading all the necessary packages.

```
library(drc)
library(survival)
library(emmeans)
# library(devtools)
# install_github("OnofriAndreaPG/AgriCensData")
library(agriCensData)
data(Germination)
head(Germination)
```

```
##   obsT temp counts
## 1    1   15      2
## 2    3   15     12
## 3    5   15     11
## 4    7   15     13
## 5    9   15     14
## 6   11   15      4
```

The dataset reflects the way the data were collected: the data are shown as counts (`counts`) of germinated seeds at each assessment time (`obsT`) and temperature (`temp`). Note that it is not true that, for instance, 12 seeds germinated 3 days After the Beginning of the Assay (ABA) (look at the second raw of the dataset): actually, they germinated in an unknown moment between 1 and 3 ABA.

A traditional nonlinear regression model

The traditional method of data analysis is based on

1. transforming the observed counts into cumulative proportions of germinated seeds and
2. fitting a Gaussian cumulative probability density function, using nonlinear least squares regression.

We accomplished this part by using the `drm()` function in the `drc` package (Ritz et al. 2015). We decided to fit a log-normal (`LN.2()`) probability density function, because we expected the distribution of germination times within the population to be log-normal. The argument `curveid` specifies that we need two curves (one per temperature), and the argument `pmodels` specify that the first parameter (b, a slope of the curve in the inflection point) is common for the two curves (~1), while the second (e, time to 50% germination, i.e., T50) depends on the temperature.

```r
cumProp <- as.numeric(
  unlist(with(
    Germination,
    tapply(counts, temp, cumsum)))/100
  )

mod <- drm(cumProp ~ obsT, fct = LN.2(),
           curveid = temp,
           pmodels = list(~ 1, ~ temp - 1),
           data = Germination)
summary(mod)
```

```
##
## Model fitted: Log-normal with lower limit at 0 and upper limit at 1 (2 parms)
##
## Parameter estimates:
##
##                Estimate Std. Error t-value   p-value
## b:(Intercept) 0.871778   0.017726  49.182 < 2.2e-16 ***
## e:temp15      9.106622   0.178919  50.898 < 2.2e-16 ***
## e:temp25      5.429718   0.136514  39.774 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error:
##
##  0.02043963 (37 degrees of freedom)
```

As noted in the paper (*Literature reference, when available*), this traditional method of data analysis is suboptimal for several reasons:

1. modelling cumulative proportions instead of observed counts does not match the real process of data collection
2. nonlinear regression assumes that a certain proportion of germinated seeds is reached at the exact moment of observation, but this does not have to be true
3. nonlinear regression assumes model errors (deviations of observed proportions from the fitted curve) to be normally distributed, homoscedastic, and independent, which assumptions are often violated in germination assays.

A time-to-event model

For this example, a time-to-event model is more appropriate than nonlinear regression. The term *time-to-event model* is just another name for a survival model used when we are studying the time to the occurence of an event that is not the death.

In order to fit a time-to-event model in R, we could use the `survreg()` function in the `survival` package (Therneau 1999), as we did for the first example. The observational unit is the seed and, therefore, we need to reorganise the dataset, so that there is one record for each seed in the lot (200 records in all). For each temperature subset, records are added also for the seeds that did not germinate at the end of the assay. Each record needs to be characterised by three variables: the temperature at which the seed was tested and the two limits of the interval within which it germinated (`timeBef` and `timeAf`). Seeds which already germinated at the first monitoring time have `timeBef` equal to NA (not available). Seeds which did not germinate at the final monitoring time have `timeAf` equal to NA. The process of data reorganisation can be done by using the following code, which produces the dataset `germination2`.

```
countsC <- with(
  Germination,
  c(counts[1:20],
    100 - sum(counts[1:20]),
    counts[21:40],
    100 - sum(counts[21:40]))
  )

Germination2 <- data.frame(
  temp = rep(factor(rep(c(15, 20), each = 21)), countsC),
  timeBef = rep(c(NA, Germination$obsT[1:20], NA,
                  Germination$obsT[21:40]), countsC),
  timeAf = rep(c(Germination$obsT[1:20], NA,
                 Germination$obsT[21:40], NA), countsC)
  )

rm(countsC)
head(Germination2, 10)
```

```
##    temp timeBef timeAf
## 1    15      NA      1
## 2    15      NA      1
## 3    15       1      3
## 4    15       1      3
## 5    15       1      3
## 6    15       1      3
## 7    15       1      3
## 8    15       1      3
## 9    15       1      3
## 10   15       1      3
```

Now we can fit a time-to-event model, using a log-normal distribution of germination times (`dist = "lognormal"`).

```
modTE1 <- survreg(Surv(timeBef, timeAf, type = "interval2") ~ temp,
  dist = "lognormal", data = Germination2)
summary(modTE1)
```

```
##
## Call:
```

```
## survreg(formula = Surv(timeBef, timeAf, type = "interval2") ~
##     temp, data = Germination2, dist = "lognormal")
##               Value Std. Error     z       p
## (Intercept)  2.2415     0.1173 19.11 < 2e-16
## temp20      -0.6221     0.1666 -3.73 0.00019
## Log(scale)   0.1453     0.0563  2.58 0.00989
##
## Scale= 1.16
##
## Log Normal distribution
## Loglik(model)= -525.3   Loglik(intercept only)= -532.1
##  Chisq= 13.57 on 1 degrees of freedom, p= 0.00023
## Number of Newton-Raphson Iterations: 3
## n= 200
```

This function returns three parameters: (i) the logarithm of the time to 50% germination (T50) at 15°C (Intercept), (ii) the difference in log-T50 between 20 and 15°C, and (iii) the standard deviation of the log-normal distribution of germination times, assumed to be unaffected by temperature (homoscedasticity). We may be interested in the T50 at both temperatures. We can retrive them by back-transforming the model parameters (e.g $exp(2.2415)$ and $exp(2.2415 - 0.6221)$), which is easily done by using the `emmeans()` function in the `emmeans` package (Lenth 2016).

```
emmeans(modTE1, ~ temp, transform = "response")
```

```
##  temp response        SE  df lower.CL  upper.CL
##  15   9.407324 1.1034551 197 7.231223 11.583425
##  20   5.050110 0.5974451 197 3.871901  6.228319
##
## Confidence level used: 0.95
```

A more general method to fit time-to-event models

The `survreg()` function is a good tool to fit time-to-event models, but it has a notable limitation: it assumes that all individuals should experience the event of interest in one specific time, either before or after the end of the experiment. While this may be acceptable with survival studies, in germination assays there will almost always be a final fraction of dormant seeds, which will never germinate at the given environmental conditions. In our example, this fraction was negligible and, therefore, `survreg()` gave a good fit.

In general, with seed germination assays, it is necessary to use a time-to-event function which accounts for the final fraction of germinated seeds. In R, we can use the `drm()` function in the `drc` package. This function works with datasets in the grouped form, like the original dataset `Germination`, which we used for nonlinear regression. However, for each Petri dish, we need to add one record to store the number of ungerminated seeds at the end of the assay. For each record, we should specify the temperature, the number of germinated seeds, and the two limits of the interval within which those seeds germinated. For the ungerminated seeds, the lower limit of the interval is set to the final assessment time, and the highest limit is set to `Inf` (see below).

The model call is similar to the one we used for nonlinear regression, with the only differences in the dependent variable (the limits of the interval the germinations took place within) and the `type` argument, which has been explicitely set to `event`.

```
countsC <- with(Germination,
                c(counts[1:20], 100 - sum(counts[1:20]),
                  counts[21:40], 100 - sum(counts[21:40]))
                )
```

```r
Germination3 <- data.frame(
  temp = factor(rep(c(15, 20), each = 21)),
  timeBef = c(0, Germination$obsT[1:20],
              0, Germination$obsT[21:40]),
  timeAf = c(Germination$obsT[1:20], Inf,
             Germination$obsT[21:40], Inf),
  counts = countsC)

head(Germination3, 10)
```

```
##    temp timeBef timeAf counts
## 1    15       0      1      2
## 2    15       1      3     12
## 3    15       3      5     11
## 4    15       5      7     13
## 5    15       7      9     14
## 6    15       9     11      4
## 7    15      11     13      6
## 8    15      13     15      4
## 9    15      15     17      5
## 10   15      17     19      5
```

```r
modTE <- drm(counts ~ timeBef + timeAf, fct = LN.2(),
             type = "event", curveid = temp,
             data = Germination3,
             pmodels=list(~ 1, ~ temp - 1))
summary(modTE)
```

```
##
## Model fitted: Log-normal with lower limit at 0 and upper limit at 1 (2 parms)
##
## Parameter estimates:
##
##                 Estimate Std. Error t-value   p-value
## b:(Intercept) 0.864686   0.048715 17.7498 < 2.2e-16 ***
## e:temp15      9.407667   1.103586  8.5246 < 2.2e-16 ***
## e:temp20      5.050970   0.597632  8.4516 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we discuss in our paper (*Literature reference, when available*), the difference between nonlinear regression and time-to-event regression mainly relates to standard errors.

References

Lenth, RV (2016) Least-Squares Means: The *R* Package **Lsmeans**. *Journal of Statistical Software* **69**

Ritz, C, F Baty, JC Streibig, D Gerhard (2015) Dose-Response Analysis Using R. *PLOS ONE* **10**, e0146021

Therneau, T (1999) *A Package for Survival Analysis in S.* R package version 2.36-14, Survival